

FishDetectLLM: Multimodal instruction tuning with large language models for fish detection

Jiaxin Zhu ^{a,b}, Shibai Yin ^{a,b,*}, Xin Liu ^{a,b}, Xingyang Wang ^{a,b}, Yee-Hong Yang ^c

^a Department of Computing and Artificial Intelligence, Southwestern University of Finance and Economics, Chengdu, Sichuan, 611130, China

^b Kash Institute of Electronics and Information Industry, Xinjiang, 844000, China

^c Department of Computing Science, University of Alberta, Edmonton, T6G 2E8, Canada

ARTICLE INFO

Keywords:

Multimodal LLM

Fish detection

Fish classification

ChatGPT

ABSTRACT

Aquatic species play crucial roles in global ecosystems but are increasingly threatened by factors such as overfishing, coastal development and climate change. Existing deep learning methods address these challenges by employing powerful networks and large-scale, diverse datasets, separately tackling species recognition and trait identification during ongoing monitoring. However, they often exhibit limited generalization ability. Inspired by the human ability to quickly identify fish species and their locations with just a glance at an underwater image or scene, we introduce FishDetectLLM—a framework built on the lightweight TinyLLaVA architecture. FishDetectLLM utilizes the powerful reasoning capabilities and vast world knowledge of large language models (LLMs) to address the fish detection problem, providing both fish classification results and predicted bounding boxes for fish. Specifically, we create instruction dialogues for fish detection that connect fish taxonomy with classification descriptions and map location descriptions to the corresponding coordinates of bounding box in the input images from the recently released large-scale FishNet dataset. Then, we pretrain and fine-tune FishDetectLLM to achieve fish detection using the created dataset, leveraging the principle of augmenting human knowledge. Our results show that FishDetectLLM significantly outperforms existing multimodal LLMs and task-specific methods. Unlike conventional detection architectures that struggle to generalize beyond the training data, FishDetectLLM exhibits strong generalization capabilities, achieving robust performance on unseen data. This innovation paves the way for future applications of MLLMs in full research and offers valuable tools for the conservation of fish biodiversity.

1. Introduction

Since global climate change and human activities increasingly impact marine environments, aquatic biodiversity faces severe threats. Monitoring of changes in fish behavior and population dynamics is essential for maintaining ecosystem stability and health. A critical yet challenging aspect of this monitoring is fish detection, which relies on a labor-intensive and time-consuming visual data collection. Therefore, developing advanced methods for biodiversity monitoring is crucial. These methods not only aid marine biologists in tracking species populations but also facilitate real-time assessments of threats to aquatic life. Beyond biodiversity monitoring, fish detection also contribute to climate change studies by providing insights into shifts in species distribution and behavior under changing environmental conditions.

Recently, deep learning-based methods have made significant advances in the field of machine vision, also providing an efficient solution for fish detection. Existing fish detection methods can be roughly

divided into two classes: one-stage detectors and two-stage detectors. The one-stage detectors, such as the YOLO series, which predict object categories and bounding box in a single step, are the popular paradigms for fish detection. While two-stage detectors such as Fast RCNN [1], Faster-RCNN [2] and Mask-RCNN [3], use a divide-and-conquer approach, focusing on generating region proposals in the first stage and then on refining these proposals in the second stage to improve accuracy. Although existing methods achieve high accuracy in fish detection, they heavily rely on well-designed network architectures and diverse datasets (Fish4Knowledge [4] and Wild-Fish++ [5]). Additionally, exhibiting strong domain-specific properties and low generalization ability when applied to other domains.

Inspired by the human ability to quickly recognize fish locations and species from a brief glance at an underwater scene or image, we address these problems from a different perspective by introducing the FishDetectLLM model. By utilizing textual descriptions and visual cues

* Corresponding author at: Department of Computing and Artificial Intelligence, Southwestern University of Finance and Economics, Chengdu, Sichuan, 611130, China.

E-mail address: shibaiyin@swufe.edu.cn (S. Yin).

<https://doi.org/10.1016/j.knosys.2025.113418>

Received 25 September 2024; Received in revised form 10 March 2025; Accepted 22 March 2025

Available online 14 April 2025

0950-7051/© 2025 Elsevier B.V. All rights are reserved, including those for text and data mining, AI training, and similar technologies.

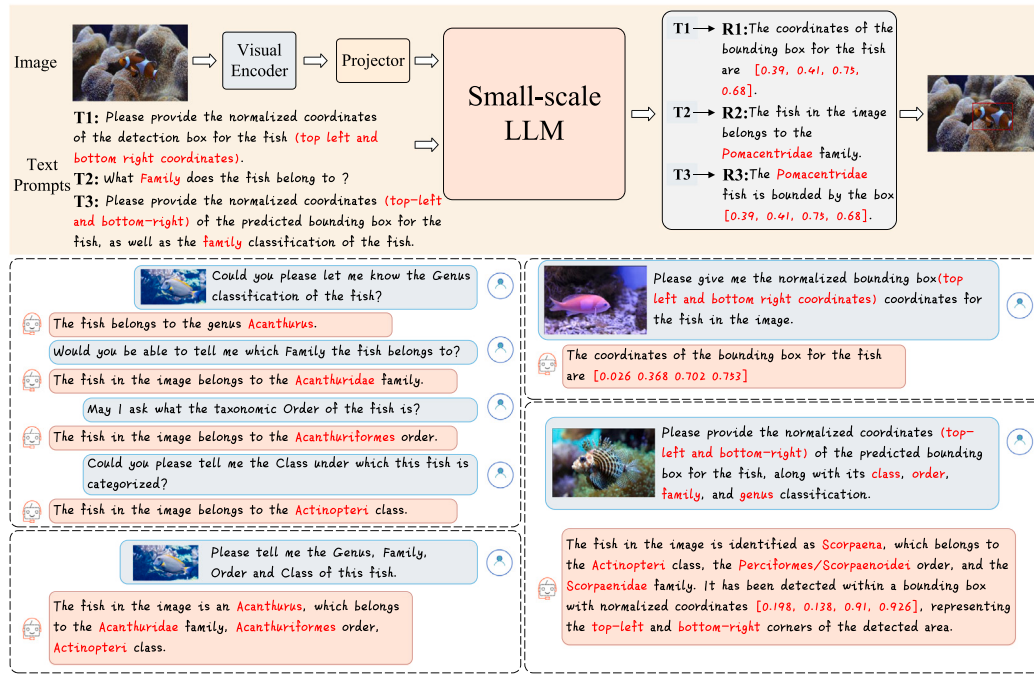


Fig. 1. We propose FishDetectLLM for performing fish detection task. By providing image and text prompts to FishDetectLLM, it obtains different responses. We showcase the scenarios: a multi-turn conversation for fish classification (top left), a single-turn conversation for fish classification (bottom left), a single-turn conversation for the bounding box prediction (top right), and a single-turn conversation for fish detection (bottom right).

through Large Language Models (LLMs), FishDetectLLM transforms detection task into a visual question-and-answer problem. To achieve this, we construct a question-answer instructional conversation dataset for fish detection by associating classification descriptions with fish taxonomy and linking fish location descriptions to the corresponding coordinates of bounding box in the input image. We then fine-tune the Multimodal Large Language Model (MLLM) using these instructional conversation datasets to obtain FishDetectLLM.

The overview of FishDetectLLM is shown in Fig. 1. As can be seen, it is built upon the lightweight MLLM named TinyLLaVA, which consists of a visual encoder, a projection layer, and a pre-trained LLM. The visual encoder learns image representations, while projection layer bridges image and text queries by aligning visual representations with text tokens. Meanwhile, the pre-trained LLM takes both image and text tokens as inputs to generate corresponding text responses.

Next, we pretrain the projector while keeping the visual encoder and LLM frozen, to better align the visual and textual information in the embedding space. Afterward, we fine-tune the entire FishDetectLLM model with the aforementioned instructional conversations, enabling it to accurately perform fish detection by outputting the fish classification results and the predicted bounding boxes for fish.

To facilitate the training and evaluate FishDetectLLM's performance, we develop instructional conversations using the recent FishNet dataset released in 2023, which features 94,532 images of aquatic species across 17,357 different species, gathered from diverse regions worldwide. Of these, 75,631 images are allocated for training and 18,901 images for testing. Since the FishNet dataset includes fish location information (coordinates of bounding boxes) and taxonomy details (Genus, Family, Order, and Class), it enables us to effectively evaluate the performance of FishDetectLLM based on fish detection. Extensive experiments show that FishDetectLLM achieves classification accuracies of 69.78%, 87.61%, 93.26%, and 99.06% at the Genus, Family, Order, and Class levels, respectively, on the FishNet testing dataset, significantly outperforming existing state-of-the-art fish classification methods. Additionally, FishDetectLLM achieves fish detection scores of 84.2 *mAP*₅₀, 80.5 *mAP*₆₀, 72.5 *mAP*₇₀, 55.5 *mAP*₈₀, and 25.5 *mAP*₉₀ on the FishNet testing dataset, demonstrating a substantial

improvement over current methods. Moreover, FishDetectLLM exhibits strong generalization ability by accurately detecting fish locations in our self-collected 110 images, which are not included in the training dataset. All the above experiments demonstrate that the FishDetectLLM model exhibits superior accuracy and generalization capabilities in fish detection.

To the best of our knowledge, FishDetectLLM is the first LLM-based model capable of fish detection, providing a crucial tool for biodiversity conservation. In summary, the key contributions of this work are as follows:

- We present FishDetectLLM, the first model to leverage a lightweight Multimodal Large Language Model (MLLM) for fish detection, harnessing the strong reasoning ability and extensive world knowledge inherited from LLMs.
- We have developed a new instruction-based conversation dataset that integrates fish classification and bounding box prediction. This integration is crucial for enabling knowledge complementarity in FishDetectLLM, thereby improving the accuracy of fish detection.
- FishDetectLLM outperforms existing methods by a large margin, demonstrating exceptional performance and strong generalization capabilities.

2. Related work

Our work spans multiple areas, including marine organism identification, computer vision and MLLM. Accordingly, we provide a brief review of fish classification and detection, considering both computer vision and MLLM perspectives.

2.1. Fish classification

Fish classification, which aims to identify and recognize fish species based on their features, categorizes the target fish into corresponding taxonomy levels, such as Genus, Family, Order, and Class, by comparing it with representative sample images. Recently, deep learning

methods have demonstrated superior performance in fish classification, and these methods can be broadly categorized into three types: ResNet (Residual Network)-based methods, ViT (Vision Transformer)-based methods, and ConvNeXt (Convolutional Next)-based methods.

ResNet-based methods rely on a ResNet-50 network for classification, which consists of 50 residual blocks and a fully connected layer. By using residual blocks to mitigate the vanishing gradient problem, ResNet-50 effectively improves classification accuracy [6]. Knausgård et al. [7] further utilize ResNet-101 for fish classification, building on the high performance achieved by ResNet-50.

ViT-based methods rely on Vision Transformers to efficiently extract global features and capture global correlations for accurate fish classification. Liu et al. [8] propose a dual-path pyramid vision transformer network for fish classification, where one path uses a ViT to extract global information, and the other path employs a CNN to capture local information. Bao et al. [9] introduce a self-supervised pre-training framework for Bidirectional Encoder Representations from Image Transformers (BEIT), achieving strong fine-tuning results in fish classification.

To further investigate the performance differences between Transformers and ResNet, Liu et al. [10] progressively modify a standard ResNet, incorporating design elements from Vision Transformers. Through this process, they identify key architectural components – collectively named ConvNeXt – that contribute to the observed performance differences. Experiments demonstrate that ConvNeXt performs competitively with Transformers in fish classification.

2.2. Fish detection

Fish detection, which aims to perform fish classification and localization tasks based on the object detectors, involves identifying fish species and their precise locations in images, facilitating a wide range of applications such as ecological monitoring, fisheries management, and environmental research. The common object detectors are one-stage detectors and two-stage detectors. One-stage detectors, such as the You Only Look Once (YOLO) series of algorithms, are the popular paradigms for object detection, due to its accuracy and efficiency. YOLOv11, focuses on an innovative feature extraction technique for enhancing the capture of fine details [11]. Ouyang et al. propose the DERT with YOLO (DEYO) model for achieving real-time object detection [12]. Additionally, Feng et al. propose a Task-aligned One-stage Object Detector (TOOD), which improves the accuracy of object detection by adjusting the outputs of classification and localization through task-interactive features [13]. Zhang et al. propose the DETR with Improved deNoising anchor boxes (DINO) which employs a contrastive approach for denoising, a dual-lookahead scheme for box prediction, and a hybrid prompt selection strategy for anchor initialization [14].

Unlike one-stage detectors, which directly predict object categories and bounding boxes in a single step, two-stage detectors use a divide-and-conquer approach, where the detection process is split into two stages: first generating region proposals, and then classifying the proposals and refining the bounding boxes. The classic two-stage detectors are Fast RCNN [15], Faster RCNN [2] and Mask RCNN [3].

Specifically, Ren et al. improve the speed and performance of RCNN by proposing Faster RCNN, which introduces a Region Proposal Network (RPN) for generating candidate regions more efficiently [2]. He et al. further propose Mask RCNN, which extends Faster RCNN by adding a segmentation mask branch to the network, enabling precise pixel-level segmentation along with object detection [3]. Hou et al. propose Saliency DERT, a framework designed to address the encoding and selection redundancies prevalent in two-stage DETR-like detectors [16].

Although deep learning-based object detectors achieve promising performance, they exhibit domain-specific properties when trained on large-scale, diverse datasets like Fish4Knowledge and WildFish++. This domain specificity can lead to performance drops due to domain shifts, resulting in reduced generalization ability.

2.3. Multimodal large language model

Inspired by the powerful reasoning abilities of large language models (LLMs) in natural language understanding tasks, several open-source LLMs, such as Vicuna, LLaMA, and Alpaca, have been released, allowing researchers to explore their potential in multimodal tasks. Li et al. [17] propose a lightweight transformer called the Querying Transformer to extract visual features from a frozen image encoder and then feed the most relevant features into the LLM to generate the desired text. Liu et al. [18] propose the Large Language and Vision Assistant (LLaVA), which integrates a vision encoder with a large language model (LLM) to enable general-purpose visual and language understanding. Despite the success of MLLMs in multimodal tasks, their effectiveness in purely vision tasks remains largely unexplored. Wang et al. [19] propose LocLLM, which depends on the strong reasoning ability of LLMs and visual clues for human keypoint detection. Feng et al. [20] employ the powerful capabilities of MLLMs to generate speculative poses and reason about pose estimation. However, none of the above works explore the potential of MLLMs in marine population monitoring. To the best of our knowledge, FishDetectLLM is the first to leverage MLLMs for fish detection.

3. Method

3.1. Overview

While traditional methods have advanced fish species recognition and identification, their reliance on fixed labels defined in the training dataset limits their ability to generalize effectively to diverse and unseen scenarios. Inspired by the human ability to quickly and intuitively identify fish species and their locations in underwater environments at a glance, we aim to address these limitations by leveraging the world knowledge and reasoning capabilities of MLLMs to perform fish detection. FishDetectLLM bridges the gap between the limitations of current methods and the broader demands of biodiversity conservation, providing a foundation for sustainable ecosystem management and unlocking expanded applications of MLLMs.

Concretely, FishDetectLLM takes an fish image and a text prompt as input, and outputs three different responses as follow:

$$F_{LLM}(I, T_i) = \begin{cases} \{l_1, l_2, l_3, l_4\} & \text{if } i = 1, \\ \{f_g, f_f, f_o, f_c\} & \text{if } i = 2, \\ \{l_1, l_2, l_3, l_4\}, \{f_g, f_f, f_o, f_c\} & \text{if } i = 3. \end{cases} \quad (1)$$

where F_{LLM} denotes the model of FishDetectLLM. T_i represent a text prompt corresponding to different vision tasks indexed by i . When $i = 1$, T_1 is the prompt for predicting bounding box. Hence, FishDetectLLM takes the image I and detection prompt T_1 as input, outputting the text sequence $\{l_1, l_2, l_3, l_4\}$ which represents the coordinates of the bounding box for detected fish. When $i = 2$, T_2 prompts fish classification. Given I and T_2 , FishDetectLLM outputs the text sequence $\{f_g, f_f, f_o, f_c\}$ for fish classification, where f_g denotes the genus, f_f the family, f_o the order, and f_c marks the species. When $i = 3$, T_3 prompts both fish classification and bounding box prediction. FishDetectLLM processes both I and T_3 , jointly outputting both classification and detection results. This approach allows FishDetectLLM to treat fish detection as a visual question-answering task, leveraging a multimodal framework to generate text-based responses.

To do so, we build FishDetectLLM based on the TinyLLaVA with only 3.0B parameters. As displayed in Fig. 2, FishDetectLLM contains three main components: a visual encoder V_ϕ , a projector module P_ϕ , and a small-scale LLM L_θ , where θ , ϕ , and Φ represent the learnable parameters. Then $F_{LLM}(I, T_i)$ in Eq. (1) can be rewritten as $L_\theta(P_\phi(V_\phi(I)), T_i)$.

The vision encoder V_ϕ takes the image I as input and generates a sequence of visual patch features $\{v_j\}_{j=1}^M$ via $V_\phi(I)$, where M is the total

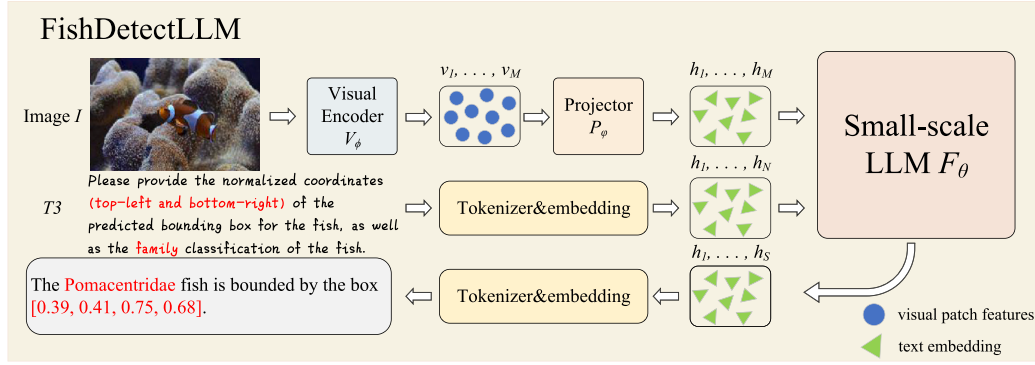


Fig. 2. The proposed FishDetectLLM for fish detection utilizes the TinyLLaVA model, which consists of a visual encoder V_ϕ , a projector P_ϕ and a small-scale LLM L_θ . The image is processed by the visual encoder, following the projector to extract image tokens. The LLM takes both text tokens and image tokens as inputs and outputs the coordinates of the predicted bounding box.

number of visual patch features. All these visual patch sequences are further mapped into the text embedding space $\{h_j\}_{j=1}^M$ by $P_\phi(\{v_j\}_{j=1}^M)$. The small-scale LLM L_θ processes text prompt T_i using the embedding module of the LLM and generates text feature in the embedding space $\{h_k\}_{k=1}^N$, where N represents the dimension of the text embedding space. Then, the attention mechanism of the LLM allows it to focus on the inherent relationships between the visual and text embedding spaces, generating the output sequence with the help of the LLM's tokenizer:

$$L_\theta(v_1, \dots, v_M, h_1, \dots, h_N) = \{r_1, \dots, r_s, \dots, r_S\} \quad (2)$$

where r_s represents output of Eq. (1) indexed by i . It is noteworthy that r_s is generated based on previous information $\{v_1, \dots, v_M, h_1, \dots, h_N, r_1, \dots, r_{s-1}\}$.

3.2. Training and instructional conversation

We use the multi-round conversation strategy to boost training efficiency. This approach allows us to obtain both classification results and bounding box prediction results from a single image. Therefore, the data for training FishDetectLLM contains image-text pairs (I, R) , where R is generated using a multi-turn conversation of the form $R = (T_1^1, R_a^1, \dots, T_i^n, R_a^n, \dots, T_i^N, R_a^N)$, with N representing the number of conversation rounds. T_i^n and R_a^n are the prompt text and the corresponding response at the n th conversation round, respectively. The training process contains: pre-training and fine-tuning.

3.2.1. Pre-training and fine-tuning

Although the framework based on Tiny LLaVA contains only 3.0B parameters, it is still not feasible to train the entire FishDetectLLM model with limited GPU resources. Hence, we train FishDetectLLM in two stages: a pre-training phase and a fine-tuning phase. In the pre-training phase, we freeze the parameters of both the visual encoder V_ϕ and the small-scale LLM L_θ , focusing solely on training the projector P_ϕ to better align the visual and textual information in the embedding space. To do so, we feed image-caption style data (I, R_a) from the LLaVA-1.5-558k dataset [21] into FishDetectLLM and obtain the output text sequence $(r_1, \dots, r_s, \dots, r_S)$. It is noteworthy that I is the image and R_a is one of the responses from R . Then, the sequence $(r_1, \dots, r_s, \dots, r_S)$ is further mapped to the LLM vocabulary via a linear classifier l . On the other hand, the ground truth R_a is also encoded into the vocabulary sequence $(r_1^*, \dots, r_s^*, \dots, r_S^*)$. The FishDetectLLM model is optimized using the cross-entropy loss function:

$$L = \sum_{s=1}^S CE(l(r_s), r_s^*) \quad (3)$$

During the fine-tuning phase, we train all components of FishDetectLLM using the image-text and text-text pairs from (I, R) provided by the

created Fine-tuning dataset and the LLaVA-1.5-mix-665k dataset [21], where R represents the complete multi-round conversation. We optimize the parameters of FishDetectLLM with the loss function L .

3.2.2. Instructional conversation

Constructing appropriate instructional conversation is crucial for effectively training FishDetectLLM. To enable FishDetectLLM to classify the target fish into Genus, Family, Order, and Class and to accurately detect its location, we develop both fish classification-based and bounding box prediction-based instructional conversations. These conversations are based on the recent FishNet dataset, the largest and most diverse fish dataset currently available. FishNet comprises 94,532 images from 17,357 aquatic species, categorized by fish taxonomy (Genus, Family, Order, and Class), facilitating the creation of robust classification-based instructional conversations. Besides, the dataset includes bounding box annotations, which are instrumental in building bounding box prediction-based instructional conversations. Specifically, we create fish classification-based instructional conversations using image-text pairs (I, R) by formatting the FishNet data into a question-and-answer format. The template is shown as follows:

"HUMAN": <IMAGE> Can you tell me which Genus/Family/Order Class this fish belongs to?

"ASSISTANT": Sure, this fish belongs to <Genus/Family/Order/Class> classification.

Additionally, to improve efficiency, we also design a comprehensive template that provides all classification information for one-round conversation:

"HUMAN": <IMAGE> Could you please provide the Genus, Family Order, and Class information for this fish?

"ASSISTANT": The fish is classified under the Genus <genus>, Family <family>, Order <order>, and Class <class>.

For bounding box prediction, the key point in constructing the instructional conversation is determining the optimal format for the coordinates of the bounding box, enabling FishDetectLLM to improve detection accuracy. A straightforward approach is to use the pixel locations of the top-left and bottom-right corners of the detected bounding box. However, this strategy suffers from low precision due to the inconsistent image scale in the FishNet dataset. Motivated by the work of Wang et al. [19], we use the decimal string approach to represent the coordinates of the bounding box by normalizing the pixel locations of the top-left and bottom-right corners relative to the image size. Therefore, we construct the instructional conversation following this format:

"HUMAN": <IMAGE> Can you provide the normalized coordinates of the bounding box for detected fish?

"ASSISTANT": Sure, the normalized coordinates of the bounding box for the detected fish are as follows: <location>.

Table 1

Ablation study on the composition of instruction conversation on the FishNet Testing dataset. The best results are highlighted in bold.

Fish classification				Bounding box Prediction				
IT2C+LLaVA-1.5-mix-665k				IT2P+LLaVA-1.5-mix-665k				
Genus	Family	Order	Class	<i>mAP</i> 50	<i>mAP</i> 60	<i>mAP</i> 70	<i>mAP</i> 80	<i>mAP</i> 90
67.14%	86.41%	93.38%	99.09%	82.63	77.73	71.33	54.21	23.42
IT2C+IT2P+LLaVA-1.5-mix-665k				IT2C+IT2P+LLaVA-1.5-mix-665k				
Genus	Family	Order	Class	<i>mAP</i> 50	<i>mAP</i> 60	<i>mAP</i> 70	<i>mAP</i> 80	<i>mAP</i> 90
69.78%	87.61%	93.40%	99.09%	84.23	80.48	72.25	55.45	25.45

Table 2

Ablation study on the representation of instruction conversation for bounding box prediction on the FishNet Testing dataset. The best results are highlighted in bold.

	<i>mAP</i> 50	<i>mAP</i> 60	<i>mAP</i> 70	<i>mAP</i> 80	<i>mAP</i> 90
PC	42.91	27.28	15.59	6.84	1.25
NPL	84.23	84.48	72.25	55.45	25.45

<location> represents the normalized coordinates rounded to three decimal places within the range [0,1], i.e., [0.abc,0.def,0.ghi,0.jkl] and the lowercase letters such as $a \sim i$ represent any digit from 0 to 9. It is noteworthy that the tokenizer of the LLM will decompose a decimal string into separate tokens. For example, 0.abc will be divided into the tokens {"0", ".", "a", "b", "c"}, each of which is part of the LLM's vocabulary.

4. Experiments

To build FishDetectLLM, we employ TinyLLaVA [22] as the MLLM framework, with SigLIP [23] as the visual encoder, StableLM-2-1.6B as the pre-trained LLM, and a two-layer multilayer perceptron with GELU activation as the projector. In the pre-training stage, we lock the SigLIP encoder and the StableLM-2-1.6B LLM [24], and train the projector from scratch for achieving better alignment of visual and textual information in the embedding space. Later, we further fine-tune the entire FishDetectLLM model for specific fish classification and bounding box prediction tasks. The related datasets, evaluation metrics, ablation studies, and experimental results are reported in this section.

4.1. Datasets and evaluation metrics

We conduct experiments on various datasets, including the large-scale FishNet dataset for aquatic organisms [25], as well as datasets, such as the LLaVA-1.5-558k dataset [21] and the LLaVA-1.5-mix-665k dataset [26] provided by LLaVA.

FishNet is a diverse dataset comprising 94,532 organized images representing 17,357 aquatic species. Of these, 75,631 images are allocated for the **FishNet Fine-tuning dataset**, and 18,901 images are designated for the **FishNet Testing dataset**. FishNet is organized according to aquatic biological taxonomy, including Genus, Family, Order and Class, and is also supplemented with manually annotated bounding boxes, making it suitable for both fish classification and bounding box prediction tasks. We utilize the FishNet dataset to generate instructional conversations for fish detection, which are then used to fine-tune FishDetectLLM. For clarity, the image-text instructional conversations for classification are named as IT2C, while the image-text pairs for bounding box prediction are referred to as IT2P.

LLaVA-1.5-558k, constructed by LLaVA [21], contains 558,000 image-text pairs. By pairing textual descriptions with the corresponding visual data, LLaVA-1.5-558k enables MLLM to handle complex prompts that involve both visual and textual reasoning. We use this dataset for pre-training FishDetectLLM.

LLaVA-1.5-mix-665k is an extended and enhanced version of the LLaVA series [21], consisting of 665,000 image-text pairs. To further enhance FishDetectLLM's capabilities in fish detection, we use this

dataset to fine-tune FishDetectLLM for a better alignment between text and visual information.

We use commonly accepted evaluation metrics to assess FishDetectLLM's performance in fish classification and bounding box prediction. Specifically, we apply *mAP* (mean Average Precision) to evaluate the accuracy of the bounding box prediction task. By setting the Intersection over Union (IoU) threshold at 50%, 60%, 70%, 80%, and 90%, the *mAP* metric measures FishDetectLLM's ability to accurately predict the fish location by quantifying the overlap between the predicted and the ground truth bounding boxes. Additionally, we adopt the average classification accuracy to evaluate FishDetectLLM's performance in correctly identifying the fish's Genus, Family, Order, and Class.

4.2. Implementation details

Our implementation is in PyTorch and runs on 8 NVIDIA RTX A800 GPUs, each with 80 GB of memory, utilizing the DeepSpeed engine. During the pre-training phase, we train only the projector layer on the LLaVA-1.5-558k dataset for one epoch while keeping the rest of the FishDetectLLM components frozen. The learning rate is set to 10^{-3} and the batch size is configured to 256. In the fine-tuning phase, we train the entire FishDetectLLM model, including the visual encoder, the small-scale LLM, and the projector, on the LLaVA-1.5-mix-665k dataset and the FishNet training set for one epoch, using a learning rate of 2×10^{-5} and a batch size of 128.

4.3. Ablation study

4.3.1. Effectiveness of instructional conversation

To enhance FishDetectLLM's reasoning and understanding through complementary knowledge, multi-turn conversation R incorporates both IT2C and IT2P. For evaluating composition, we fine-tune FishDetectLLM separately with each dataset (IT2C or IT2P) and LLaVA-1.5-mix-665k. Table 1 presents the quantitative results, showing that combining both IT2C and IT2P with LLaVA-1.5-mix-665k for fine-tuning FishDetectLLM significantly improves classification and bounding box prediction accuracy compared to training on a single dataset.

We also explore the effectiveness of bounding box representation for fish detection. As discussed in Section 3.2.2, we use the decimal string approach to represent the coordinates of the bounding box by normalizing the pixel locations (NPL) of the top-left and bottom-right corners relative to the image size. An alternative method for representing the location of a bounding box is to use pixel coordinates (PC). To compare the effect of different bounding box representation (NPL and PC), we use both representations to construct IT2P. Table 2 presents the differences in detection performance when fine-tuning FishDetectLLM using different representations of IT2P. As shown, using pixel coordinates to position the bounding box causes FishDetectLLM to struggle with accurately predicting the bounding box, which may be due to varying image scales.

Table 3

Ablation study on the effectiveness of the fine-tuning strategy on the FishNet Testing dataset. The best results are highlighted in bold.

V_ϕ	P_ϕ	L_θ	Gneus	Family	Order	Class	$mAP50$	$mAP60$	$mAP70$	$mAP80$	$mAP90$
	✓		29.52	61.19	75.69	95.20	53.62	42.15	30.16	17.42	5.23
✓	✓		55.45	80.21	90.33	98.79	74.46	66.02	51.38	31.09	8.09
	✓	✓	63.29	81.46	89.29	98.82	81.70	76.46	65.74	45.36	15.61
✓	✓	✓	69.78	87.61	93.40	99.09	84.23	80.48	72.25	55.45	25.45

Table 4

Ablation study on performance differences of FishDetectLLM among various LLMs on the FishNet Testing dataset. The best results are highlighted in bold.

LLM	Fish Classification				Bounding Box Prediction				
	Genus	Family	Order	Class	<i>mAP</i> 50	<i>mAP</i> 60	<i>mAP</i> 70	<i>mAP</i> 80	<i>mAP</i> 90
Phi-2	67.73%	86.94%	93.35%	99.07%	82.83	78.12	68.37	48.49	17.73
TinyLlama	69.57%	87.46%	92.82%	99.09%	83.95	79.86	71.19	54.03	24.02
StableLM-2	69.78%	87.61%	93.40%	99.09%	84.23	80.48	72.25	55.45	25.45

Table 5

Test performance of various methods on the FishNet Testing dataset, categorized by common, “FL” denotes the use of focal loss during training, while “CB” represents class-balanced training. The best results are highlighted in bold.

Methods	Family Level				Order Level			
	Common	Medium	Rare	All	Common	Medium	Rare	All
ResNet-34 [30]	77.16%	69.10%	36.65%	40.82%	83.42%	77.87%	47.36%	52.07%
ResNet-50 [30]	76.82%	70.27%	35.99%	40.37%	82.46%	75.93%	42.84%	47.94%
ResNet-101 [30]	75.73%	69.00%	31.61%	36.38%	81.35%	75.01%	42.32%	47.36%
ViT-S [31]	76.08%	67.02%	33.67%	37.97%	84.14%	74.78%	44.09%	48.79%
ViT-B [31]	82.93%	74.50%	38.26%	42.91%	88.52%	81.64%	52.44%	56.93%
ViT-L [31]	85.51%	77.05%	44.18%	48.40%	89.02%	83.89%	55.94%	60.26%
BeiT [9]	86.09%	77.67%	50.78%	54.26%	91.41%	88.24%	38.16%	45.97%
ConvNeXt* [10]	42.63%	26.56%	12.57%	14.53%	63.58%	40.67%	17.82%	17.82%
ConvNeXt [10]	90.32%	85.13%	57.03%	60.61%	94.07%	90.49%	64.84%	68.81%
ConvNeXt [10]+FL [32]	88.28%	82.02%	51.22%	55.16%	87.60%	81.11%	22.22%	31.37%
ConvNeXt [10]+CB [33]	90.53%	84.80%	57.94%	61.38%	93.15%	90.56%	71.41%	74.38%
ConvNeXt [10]+FL [32]+CB [33]	84.89%	77.92%	48.99%	52.71%	85.14%	57.25%	39.28%	41.76%
Faster-RCNN [2]	45.55%	7.04%	1.19%	15.46%	51.33%	17.92%	6.32%	23.61%
Mask-RCNN [3]	80.37%	33.18%	13.67%	40.44%	87.59%	42.88%	21.41%	49.02%
LLaVA [34]	86.85%	85.31%	61.46%	78.75%	89.81%	88.88%	77.24%	86.17%
CLIP [35]	87.99%	74.65%	54.59%	59.94%	94.44%	82.55%	69.77%	69.50%
DINO [14]	10.37%	8.73%	2.46%	7.45%	45.08%	13.06%	7.38%	16.88%
Saliency-DERT [16]	24.93%	14.78%	3.91%	14.69%	31.28%	21.39%	6.22%	20.16%
DEYO [12]	92.10%	87.28%	65.52%	83.11%	93.90%	91.30%	76.69%	88.14%
YOLOv11 [11]	81.39%	80.44%	54.35%	75.45%	92.5%	90.81%	79.63%	88.40%
FishDetectLLM	95.30%	90.49%	73.73%	87.61%	96.52%	94.31%	87.74%	93.40%

4.3.2. The effectiveness of fine-tuning strategy

The fine-tuning strategy is also crucial for the performance of MLLMs on specific vision tasks. FishDetectLLM, built on TinyLLaVA [22] with only 3.0B parameters (half the number of parameters of LLaVA-1.5’s 7B [27]), enables us to fine-tune the entire model, including a visual encoder V_ϕ , a projector module P_ϕ , and a small-scale LLM L_θ , with sufficient GPU resources. We validate the effectiveness of our fine-tuning strategy by selectively reducing the components involved in the fine-tuning phase, using the same fine-tuning datasets (FishNet and LLaVA-1.5-mix-665k). In Table 3, the components included in the fine-tuning phase are marked with a ✓. We observe fine-tuning parts of the components in FishDetectLLM leads to suboptimal results, as it fails to capture sufficient information from dataset.

4.3.3. Performance differences among various LLMs

To evaluate the performance differences of FishDetectLLM across various LLMs, we employ Phi-2 with 2.7B parameters [28] and TinyLLaVA with 1.1B parameters [29] as alternative small-scale LLMs. The results are reported in Table 4. As can be seen, StableLM-2 with 1.6B parameters, used in FishDetectLLM, achieves higher accuracy than both Phi-2 and TinyLLaVA, despite Phi-2 having more parameters than StableLM-2. This result demonstrates the superior generalization ability of FishDetectLLM and also suggests that its effectiveness depends on the capabilities of small-scale LLMs.

4.4. Fish classification

In this section, we verify the performance of FishDetectLLM on fish classification tasks, focusing particularly on the classification results at the Family and Order levels. For each level, we further partition the FishNet dataset into 3 categories based on FishNet’s long-tailed distribution for classification: 25% of the total images are designated as common categories, 50% as medium categories, and the remaining 25% as rare categories, resulting in 6 common, 52 medium, and 405 rare categories. Existing methods encompass not only standalone fish classification techniques, such as ResNet-based methods (e.g., ResNet-34 [30], ResNet-50 [30], ResNet-101 [30]), ViT-based methods (e.g., ViT-S [31], ViT-B [31], ViT-L [31]), BeiT [9], and ConvNext [10], but also include the state-of-the-art object detection methods which utilize object classification as an intermediate step, such as Faster-RCNN [2], Mask-RCNN [3], Saliency-DERT [16], DINO [14], DEYO [12] and YOLOv11 [11]. Furthermore, considering that FishDetectLLM is an MLLM-based approach, leveraging both visual and textual cues for fish classification, the MLLM-based models, such as LLaVA-V1.3-7B [34] and CLIP [35], are also selected for comparison. To be fair, we pretrain all compared methods, except LLaVA-V1.3-7B [34] and CLIP [35], on the FishNet fine-tuning dataset. For LLaVA-V1.3-7B and CLIP, we pretrain and fine-tune them with the same datasets as FishDetectLLM. Table 5 shows that FishDetectLLM achieves the highest accuracy across all categories when compared with the other methods. Furthermore,

Table 6

Test performance of various methods on the FishNet testing dataset, categorized by common, medium, and rare classes at the order and family levels. The best results are highlighted in bold.

Level	Method	Common/Medium/Rare				
		mAP50	mAP60	mAP70	mAP80	mAP90
Family	YOLOF [36]	67.2/53.1/27.1	64.0/50.4/26.6	54.9/43.3/24.4	35.5/27.3/18.8	9.9/7.9/7.7
	TOOD [37]	81.1/64.0/22.5	77.1/60.1/21.9	66.5/51.2/20.1	44.0/32.8/15.7	14.5/10.9/7.7
	Faster-RCNN [2]	39.8/28.8/12.3	34.1/23.2/8.8	23.4/18.9/5.4	9.9/6.2/4.3	2.3/1.2/0.5
	Mask-RCNN [3]	52.3/42.3/12.7	39.8/37.9/10.4	25.4/19.4/8.3	10.3/7.4/5.4	2.7/3.2/1.2
	DINO [14]	66.4/35.6/9.1	62.0/33.4/8.8	51.5/28.2/8.0	33.0/18.3/6.3	13.2/10.9/4.4
	Saliency-DERT [16]	61.8/34.2/10.9	58.0/31.9/10.4	49.6/27.1/9.5	32.9/18.1/7.6	9.0/5.8/3.2
	DEYO [12]	77.3/75.6/50.6	72.8/71.3/49.4	61.2/60.6/45.7	47.8/39.1/35.1	14.8/11.2/15.1
	YOLOv11 [11]	78.3/59.6/29.5	73.9/55.5/28.8	62.4/46.5/26.4	41.2/28.6/20.0	10.8/6.97/7.64
	LLaVA [34]	79.9/79.5/85.7	73.6/72.9/81.1	62.2/60.1/70.5	42.0/36.2/49.7	14.6/13.7/19.4
	FishDetectLLM	83.4/83.5/87.5	79.4/78.9/94.9	70.8/70.0/72.3	53.7/52.8/62.8	24.3/23.2/31.3
Order	YOLOF [36]	77.3/65.2/39.2	72.7/62.1/38.0	61.6/53.4/34.7	38.4/34.9/25.4	11.7/9.6/11.8
	TOOD [37]	84.8/76.8/50.3	80.1/73.2/48.5	69.4/64.4/43.6	46.1/42.6/32.7	15.4/14.3/13.6
	Faster-RCNN [2]	45.7/33.7/13.7	37.2/27.6/10.8	25.4/20.4/7.4	10.3/11.5/5.5	2.7/3.2/1.2
	Mask-RCNN [3]	61.2/53.1/15.3	48.9/46.7/13.7	27.4/33.1/11.4	6.3/14.7/7.6	2.1/4.3/2.8
	DINO [14]	77.8/73.7/30.0	71.7/70.6/29.3	57.3/63.0/27.0	32.1/43.4/20.9	9.32/16.74/11.3
	Saliency-DERT [16]	79.8/81.8/43.2	72.8/77.4/41.2	58.4/67.9/37.0	30.7/44.7/28.3	5.0/10.9/12.0
	DEYO [12]	83.8/89.1/61.5	76.9/85.3/59.6	61.5/74.1/53.7	34.1/50.1/37.9	7.48/15.0/14.7
	YOLOv11 [11]	85.6/91.6/76.9	79.3/87.6/75.2	64.5/77.5/68.6	38.3/54.8/52.1	9.38/18/21.9
	LLaVA [34]	84.6/90.5/91.9	76.1/84.8/86.4	60.4/73.2/75.1	36.0/51.0/53.7	9.8/19.0/22.5
	FishDetectLLM	87.3/93.3/94.0	82.5/90.0/90.9	71.2/81.9/84.1	50.9/64.2/67.9	19.7/30.1/35.2

Table 7

The performance of different methods for fish detection on the entire FishNet Testing dataset and the 110 DeepFish Testing images. 110 DeepFish images w/T represents the traditional methods trained with 990 training images, while 110 DeepFish images w/o T refers to the traditional methods not trained with 990 training images. The best results are highlighted in bold.

Method	FishNet Testing Dataset					110 DeepFish images w/T		110 DeepFish images w/o T	
	mAP50	mAP60	mAP70	mAP80	mAP90	mAP50	mAP50 – 95	mAP50	mAP50 – 95
YOLOF [36]	45.0	43.4	39.3	27.8	11.7	62.1	32.8	16.1	6.7
TOOD [37]	56.5	54.3	48.3	35.3	14.1	65.3	34.5	25.4	12.6
Faster-RCNN [2]	31.5	25.9	18.5	9.7	2.6	15.6	10.1	14.1	5.6
Mask-RCNN [3]	45.8	39.1	26.3	10.8	3.7	24.8	14.2	15.8	6.2
DINO [14]	32.9	32.0	29.2	21.7	5.4	43.7	23.2	17.7	8.4
Saliency-DERT [16]	71.2	62.3	37.4	36.3	9.4	61.4	35.7	31.4	15.3
DEYO [12]	61.4	59.4	53.2	36.9	13.3	56.7	17.2	29.7	14.2
YOLOv11 [11]	77.2	74.4	67.4	50.4	20.2	67.6	39.5	35.2	19.6
LLaVA [34]	82.1	80.3	69.9	45.0	13.4	91.3	56.2	91.3	56.2
GPT-4o [38]	70.1	65.7	55.4	37.8	16.8	40.7	10.2	40.7	10.2
FishDetectLLM	84.2	80.5	72.5	55.5	25.5	96.3	67.3	96.3	67.3

FishDetectLLM also demonstrates superior efficiency compared with MLLM-based methods. With 2 billion parameters, FishDetectLLM processes each image in 0.19 s, while CLIP (with 59 billion parameters) processes each image in 0.16 s, and LLaVA (with 7 billion parameters) takes 0.25 s.

4.5. Fish detection

We further compare FishDetectLLM with state-of-the-art object detection methods, including the one-stage detectors, such as YOLOF [36], TOOD [37], DINO [14], YOLOv11 [11], DEYO [12] and the two-stage detectors, such as Faster-RCNN [2], Mask-RCNN [3], Saliency-DERT [16]. Besides, the MLLM-based LLaVA-V1.3-7B [34] method is selected for comparison. To be specific, YOLOF [36], TOOD [37], and YOLOv11 [11] are pretrained on the FishNet fine-tuning dataset and tested on the FishNet testing dataset. While LLaVA-V1.3-7B [34] is pretrained and fine-tuned using the same datasets as FishDetectLLM. The performance of different detection methods, categorized at the Family and Order levels into common, medium, and rare categories, is presented in Table 6. Additionally, the performance of these methods, tested on the entire testing dataset without partitioning the categories, is also reported in Table 7 (see the FishNet Testing Dataset column). As shown, FishDetectLLM outperforms all the other methods not only across all categories in the testing dataset, including the rare ones, but also on the entire dataset, highlighting its strong generalization ability. More visual results are provided in Appendix B.

4.6. Evaluating generalization

To further verify the generalizability of FishDetectLLM for fish detection, we collect 1100 DeepFish images, where 990 images were used for training and the remaining 110 images for testing. The existing methods selected in Section 4.5 and GPT-4o [38] are used for comparison. To ensure fairness, we first test the performance of all methods on 110 testing images without any training or fine-tuning, in order to assess their generalization performance on images with distributions different from those in the training dataset. The results are reported in Table 7 (see the 110 DeepFish images w/o T column), where mAP50-95 represents the mean Average Precision (mAP) calculated over multiple IoU thresholds ranging from 0.50 to 0.95. As can be seen that MLLM-based methods outperform traditional methods, such as YOLOF [36], TOOD [37], Faster-RCNN [2], Mask-RCNN [3], DINO [14], Saliency-DERT [16], DEYO, and YOLOv11 [11], by a large margin, with FishLLM achieving the best performance among the MLLM-based approaches. Furthermore, we train traditional methods using 990 training images. While LLaVA-V1.3-7B [34], GPT-4o [38], and FishDetectLLM are still evaluated directly on the 110 test images in an instruction-based conversation format, without any fine-tuning. Table 7 (see the 110 DeepFish images w/T column) shows that FishDetectLLM continues to deliver the best performance on 110 DeepFish images, highlighting its strong generalization ability in fish detection.

Additionally, we select two underwater images that contain other species, such as starfish and crabs, for object detection testing. The

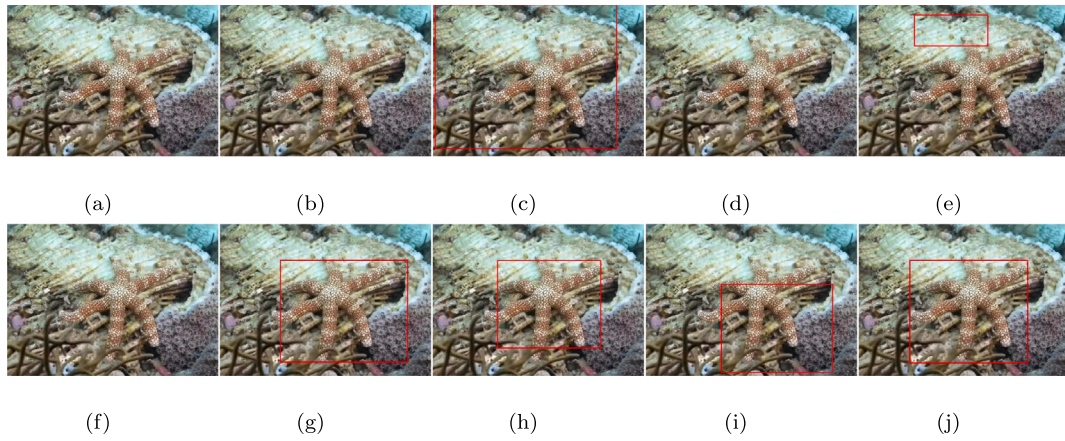


Fig. 3. Detection results on starfish image: (a) YOLOF; (b) TOOD; (c) Faster-RCNN; (d) Mask-RCNN; (e) DINO; (f) Saliency-DERT; (g) YOLOv11; (h) LLaVA; (i) GPT-4o; (j) FishDetectLLM. The red bounding box highlights detected starfish, while the image without a red bounding box shows this method fail to detect the starfish.

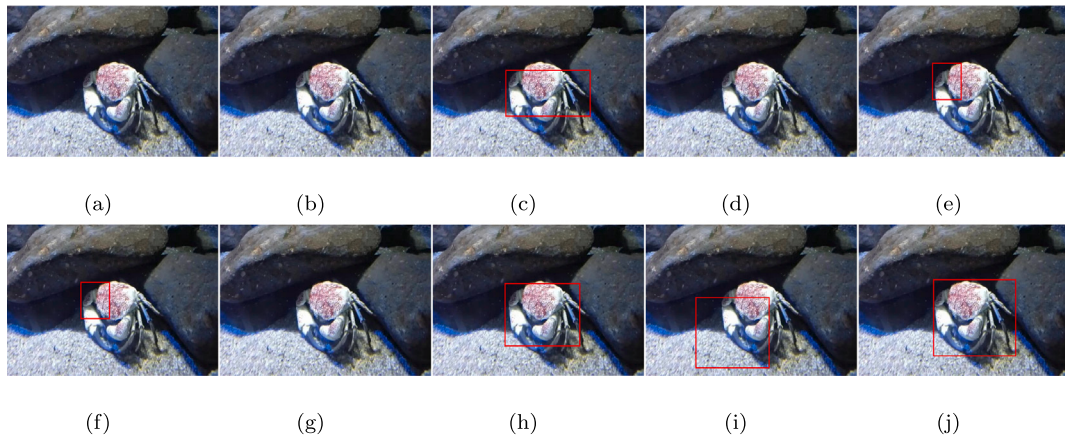


Fig. 4. Detection results on crab image: (a) YOLOF; (b) TOOD; (c) Faster-RCNN; (d) Mask-RCNN; (e) DINO; (f) Saliency-SERT; (g) YOLOv11; (h) LLaVA; (i) GPT-4o; (j) FishDetectLLM. The red bounding box highlights detected crab, while the image without a red bounding box shows this method fail to detect the crab.

visual detection results are displayed in Figs. 3 and 4. For these unseen species, traditional methods such as YOLOF [36], TOOD [37], DINO [14], YOLOv11 [11], Saliency-DERT [16], Faster-RCNN [2] and Mask-RCNN [3], fail to detect or incorrectly detect the objects. In contrast, the MLLM-based methods, such as LLaVA-V1.3-7B [34] and GPT-4o [38], are able to detect the targets, but there are issues with inaccurate bounding box prediction. For instance, LLaVA-V1.3-7B fail to include the complete starfish's legs within the bounding box (see Fig. 3(h)) and GPT-4o [38] only detects the partial starfish and the crab (see Figs. 3(i) and 4(i)). While FishDetectLLM achieves a more accurate detection of the unseen starfish and crab (see Figs. 3(j) and 4(j)), showing strong generalization ability.

5. Limitations and future works

Despite the lightweight TinyLLaVA architecture, FishDetectLLM still requires significant computational resources to perform fish detection. As a result, its inference speed may fall short of meeting the real-time requirements of practical applications such as underwater robotics, autonomous underwater vehicles (AUVs), or live monitoring systems in resource-constrained environments. Additionally, FishDetectLLM's understanding is confined to the instructions provided during pretraining and fine-tuning. For instance, it may struggle to identify complex

behavioral traits or interactions among species, which are critical for ecological research.

In future work, we will explore more efficient model compression techniques, such as model distillation, quantization or pruning, to reduce inference time and enable real-time deployment on edge devices. Furthermore, we aim to enhance FishDetectLLM's contextual understanding by incorporating multimodal data (e.g., acoustic signals, environmental metadata) and domain-specific knowledge graphs, enabling broader applications in ecological monitoring and biodiversity research.

6. Conclusion

FishDetectLLM is the first MLLM-based model for fish detection. Unlike previous approaches that rely on ad-hoc designs for various object detection architectures, FishDetectLLM leverages the reasoning capabilities and extensive world knowledge of LLMs to enhance detection performance. To achieve this, we developed instructional dialogues using the recently released FishNet dataset and fine-tuned FishDetectLLM on these instruction-based conversations. Extensive experiments demonstrate that our method significantly outperforms previous approaches and exhibits strong generalization when tested on unseen fish images with distributions differing from the training data. We hope this work inspires future research in areas such as fish trajectory tracking and beyond.

CRedit authorship contribution statement

Jiaxin Zhu: Software. **Shibai Yin:** Writing – original draft, Methodology. **Xin Liu:** Validation. **Xingyang Wang:** Validation. **Yee-Hong Yang:** Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work is supported by the Fundamental Research Funds for the Central Universities and the Natural Science Foundation of Xinjiang Uygur Autonomous Region under Grant (No. 2024D01A118), The General Project of Sichuan Provincial Philosophy and Social Science Fund under Grant (No. SCJJ24ND133), and the Sichuan Provincial Science and Technology Innovation Project under Grant (No. 25GJHZ0147).

Appendix A. Inquiry template

FishDetectLLM is designed to handle a broad spectrum of natural language inputs during the inference phase, with the following inquiry template.

Fish Classification:

"HUMAN": Could you please provide the family classification of the fish?

"HUMAN": Could you tell me which family the fish belongs to?

"HUMAN": Could you kindly inform me of the family under which this fish is classified?

"HUMAN": Would you mind sharing the family of the fish with me?

"HUMAN": May I ask what the taxonomic family of the fish is?

Bounding Box Prediction:

"HUMAN": Please detect the position of the fish in the image and return to me the coordinates of the bounding box.

"HUMAN": Give the normalized coordinates for the fish's bounding box in the image.

"HUMAN": Can you provide the normalized bounding box coordinates for the fish in the image?

"HUMAN": What are the normalized coordinates of the bounding box for the fish in the image?

"HUMAN": Please give the normalized coordinates of the bounding box around the fish in the image.

Appendix B. Visual results of fish detection

See Figs. B.5–B.8.

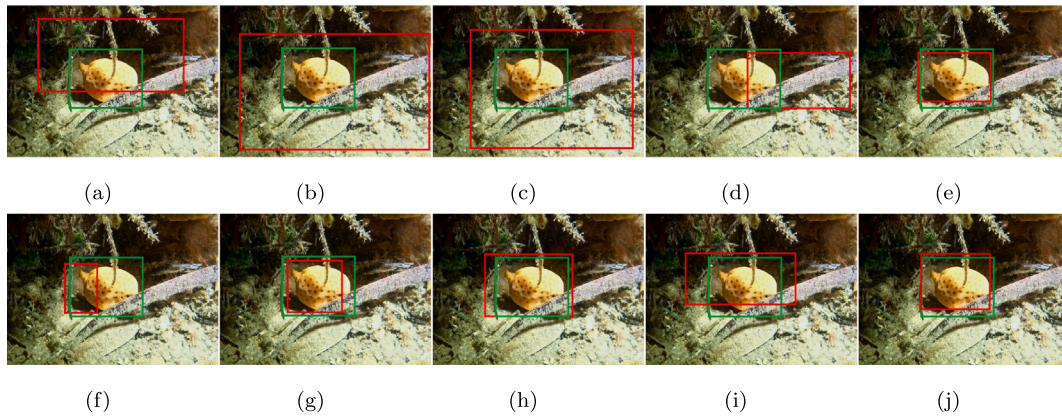


Fig. B.5. Detection results on fish image 1. The red bounding box highlight predicted fish and green bounding box highlight the ground truth: (a) YOLOF; (b) TOOD; (c) Faster-RCNN; (d) Mask-RCNN; (e) DINO; (f) Saliency-SERT; (g) YOLOv11; (h) LLaVA; (i) GPT-4o; (j) FishLLM.

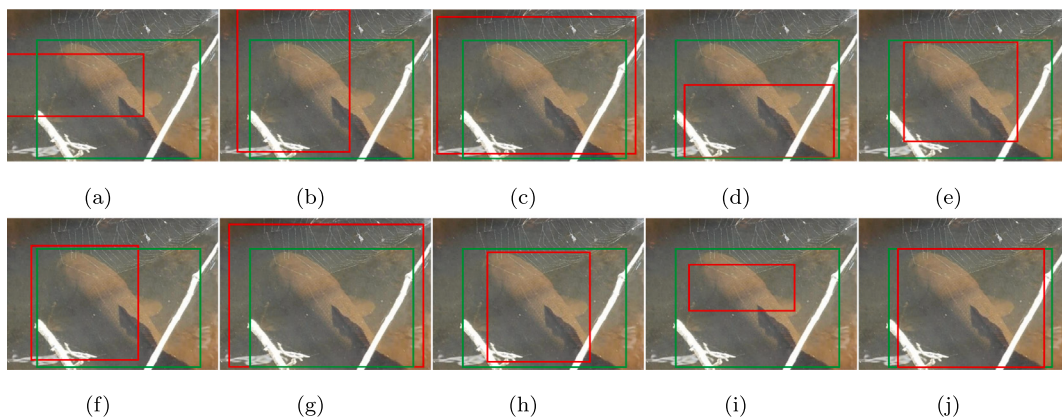


Fig. B.6. Detection results on fish image 2. The red bounding box highlight predicted fish and green bounding box highlight ground truth: (a) YOLOF; (b) TOOD; (c) Faster-RCNN; (d) Mask-RCNN; (e) DINO; (f) Saliency-SERT; (g) YOLOv11; (h) LLaVA; (i) GPT-4o; (j) FishLLM.

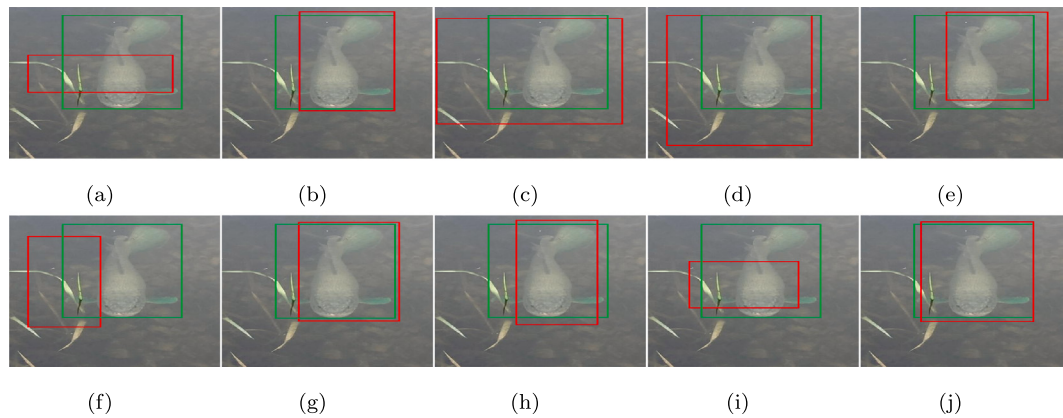


Fig. B.7. Detection results on fish image 3. The red bounding box highlight predicted fish and green bounding box highlight ground truth: (a) YOLOF; (b) TOOD; (c) Faster-RCNN; (d) Mask-RCNN; (e) DINO; (f) Saliency-SERT; (g) YOLOv11; (h) LLaVA; (i) GPT-4o; (j) FishLLM.

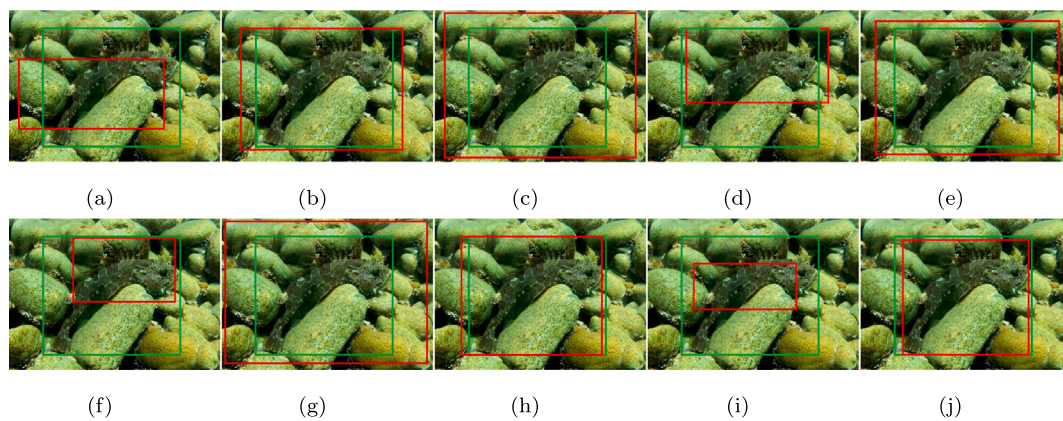


Fig. B.8. Detection results on fish image 4. The red bounding box highlight predicted fish and green bounding box highlight ground truth: (a) YOLOF; (b) TOOD; (c) Faster-RCNN; (d) Mask-RCNN; (e) DINO; (f) Saliency-SERT; (g) YOLOv11; (h) LLaVA; (i) GPT-4o; (j) FishLLM.

Data availability

Data will be made available on request.

References

- [1] K. Vijiyakumar, V. Govindasamy, V. Akila, An effective object detection and tracking using automated image annotation with inception based faster R-CNN model, *Int. J. Cogn. Comput. Eng.* 5 (2024) 343–356.
- [2] Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun, Faster R-CNN: Towards real-time object detection with region proposal networks, *IEEE Trans. Pattern Anal. Mach. Intell.* 39 (6) (2016) 1137–1149.
- [3] Kaiming He, Georgia Gkioxari, Piotr Dollár, Ross Girshick, Mask r-cnn, in: *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2961–2969.
- [4] Gayathri Nadarajan, Yun Heh Chen-Burger, Robert B. Fisher, A knowledge-based planner for processing unconstrained underwater videos, *Burger* (2009).
- [5] Peiqin Zhuang, Yali Wang, Yu Qiao, Wildfish++: A comprehensive fish benchmark for multimedia research, *IEEE Trans. Multimed.* 23 (2020) 3603–3617.
- [6] Monika Mathur, Nidhi Goel, FishResNet: Automatic fish classification approach in underwater scenario, *Sn Comput. Sci.* 2 (4) (2021) 273.
- [7] Kristian Muri Knausgård, Arne Wiklund, Tonje Knutsen Sørvalen, Kim Tallaksen Halvorsen, Alf Ring Kleiven, Lei Jiao, Morten Goodwin, Temperate fish detection and classification: a deep learning based approach, *Appl. Intell.* 52 (6) (2022) 6988–7001.
- [8] Yang Liu, Dong An, Yinjie Ren, Jian Zhao, Chi Zhang, Jiahui Cheng, Jincun Liu, Yaoguang Wei, DP-FishNet: Dual-path Pyramid Vision Transformer-based underwater fish detection network, *Expert Syst. Appl.* 238 (2024) 122018.
- [9] Hangbo Bao, Li Dong, Songhao Piao, Furu Wei, Beit: Bert pre-training of image transformers, 2021, *arXiv preprint arXiv:2106.08254*.
- [10] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, Saining Xie, A convnet for the 2020s, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 11976–11986.
- [11] Rahima Khanam, Muhammad Hussain, Yolov11: An overview of the key architectural enhancements, 2024, *arXiv preprint arXiv:2410.17725*.
- [12] Haodong Ouyang, DEYO: DETR with YOLO for end-to-end object detection, 2024, *arXiv preprint arXiv:2402.16370*.
- [13] Chengjian Feng, Yujie Zhong, Yu Gao, Matthew R Scott, Weilin Huang, Toood: Task-aligned one-stage object detection, in: *2021 IEEE/CVF International Conference on Computer Vision, ICCV*, IEEE Computer Society, 2021, pp. 3490–3499.
- [14] Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, Lionel M Ni, Heung-Yeung Shum, Dino: Detr with improved denoising anchor boxes for end-to-end object detection, 2022, *arXiv preprint arXiv:2203.03605*.
- [15] Ross Girshick, Jeff Donahue, Trevor Darrell, Jitendra Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 580–587.
- [16] Xiuquan Hou, Meiqin Liu, Senlin Zhang, Ping Wei, Badong Chen, Saliency DETR: Enhancing detection transformer with hierarchical saliency filtering refinement, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 17574–17583.
- [17] Junnan Li, Dongxu Li, Silvio Savarese, Steven Hoi, Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models, in: *International Conference on Machine Learning, PMLR*, 2023, pp. 19730–19742.
- [18] Haotian Liu, Chunyuan Li, Qingyang Wu, Yong Jae Lee, Visual instruction tuning, *Adv. Neural Inf. Process. Syst.* 36 (2024).
- [19] Dongkai Wang, Shiyu Xuan, Shiliang Zhang, Locllm: Exploiting generalizable human keypoint localization via large language model, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 614–623.
- [20] Yao Feng, Jing Lin, Sai Kumar Dwivedi, Yu Sun, Priyanka Patel, Michael J. Black, Chatpose: Chatting about 3d human pose, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 2093–2103.

- [21] Haotian Liu, Chunyuan Li, Yuheng Li, Yong Jae Lee, Improved baselines with visual instruction tuning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024, pp. 26296–26306.
- [22] Baichuan Zhou, Ying Hu, Xi Weng, Junlong Jia, Jie Luo, Xien Liu, Ji Wu, Lei Huang, Tinyllava: A framework of small-scale large multimodal models, 2024, arXiv preprint [arXiv:2402.14289](https://arxiv.org/abs/2402.14289).
- [23] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, Lucas Beyer, Sigmoid loss for language image pre-training, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023, pp. 11975–11986.
- [24] Marco Bellagente, et al., Stable lm 2 1.6 b technical report, 2024, arXiv preprint [arXiv:2402.17834](https://arxiv.org/abs/2402.17834).
- [25] Faizan Farooq Khan, Xiang Li, Andrew J. Temple, Mohamed Elhoseiny, FishNet: A large-scale dataset and benchmark for fish recognition, detection, and functional trait prediction, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023, pp. 20496–20506.
- [26] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, Devi Parikh, Making the v in vqa matter: Elevating the role of image understanding in visual question answering, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 6904–6913.
- [27] Haotian Liu, Chunyuan Li, Yuheng Li, Yong Jae Lee, Improved baselines with visual instruction tuning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024, pp. 26296–26306.
- [28] Yuanzhi Li, Sébastien Bubeck, Ronen Eldan, Allie Del Giorno, Suriya Gunasekar, Yin Tat Lee, Textbooks are all you need ii: phi-1.5 technical report, 2023, arXiv preprint [arXiv:2309.05463](https://arxiv.org/abs/2309.05463).
- [29] Peiyuan Zhang, Guangtao Zeng, Tianduo Wang, Wei Lu, Tinyllama: An open-source small language model, 2024, arXiv preprint [arXiv:2401.02385](https://arxiv.org/abs/2401.02385).
- [30] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.
- [31] Alexey Dosovitskiy, An image is worth 16x16 words: Transformers for image recognition at scale, 2020, arXiv preprint [arXiv:2010.11929](https://arxiv.org/abs/2010.11929).
- [32] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, Piotr Dollár, Focal loss for dense object detection, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 2980–2988.
- [33] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, Serge Belongie, Class-balanced loss based on effective number of samples, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 9268–9277.
- [34] Haotian Liu, Chunyuan Li, Qingyang Wu, Yong Jae Lee, Visual instruction tuning, *Adv. Neural Inf. Process. Syst.* 36 (2024).
- [35] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al., Learning transferable visual models from natural language supervision, in: International Conference on Machine Learning, PMLR, 2021, pp. 8748–8763.
- [36] Qiang Chen, Yingming Wang, Tong Yang, Xiangyu Zhang, Jian Cheng, Jian Sun, You only look one-level feature, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 13039–13048.
- [37] Chengjian Feng, Yujie Zhong, Yu Gao, Matthew R. Scott, Weilin Huang, Tood: Task-aligned one-stage object detection, in: 2021 IEEE/CVF International Conference on Computer Vision, ICCV, IEEE Computer Society, 2021, pp. 3490–3499.
- [38] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altschmidt, Sam Altman, Shyamal Anadkat, et al., Gpt-4 technical report, 2023, arXiv preprint [arXiv:2303.08774](https://arxiv.org/abs/2303.08774).